Université de Liège

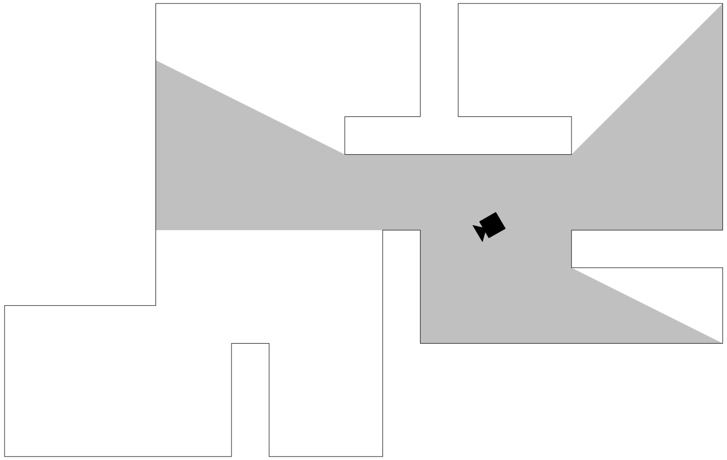
Géométrie Algorithmique

Plan du cours

- Introduction
- Arrangements dans le plan
- Triangulation de polygones
- Diagrammes de Voronoï
- Triangulation de Delaunay
- Recherche/localisation
- Arbres de partition binaire

- Exemples d'application
  - Affichage d'un polygone
    - Les cartes graphiques « savent » afficher des segments de droite, et des patchs triangulaires.
    - Toute entité plus « complexe » doit être décomposée en triangles puis affichée. C'est la cas des patchs polygonaux
    - Cette opération est faite par la carte (utilisation du GPU) ou alors par l'ordinateur (code exécuté par le CPU dans le « driver » graphique), même si l'utilisateur a l'impression que l'interface logicielle permet nativement l'affichage de polygones.
    - C'est le cas par exemple du code de visualisation choisi pour le cours...

- Exemples d'application
  - Problème de la « galerie d'art »
    - Surveiller un étage d'une galerie d'art avec un nombre adéquat de caméras positionnées convenablement

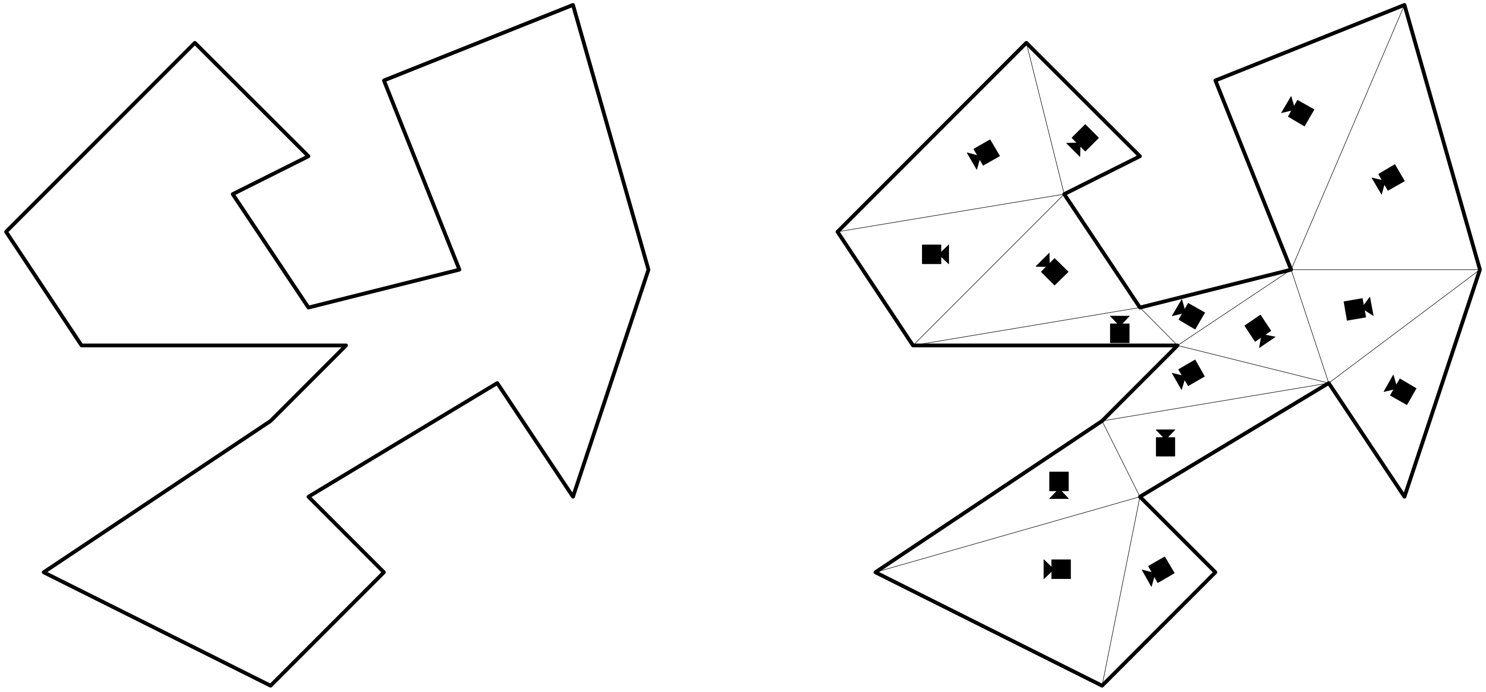


Triangulation de polygones

- Lien entre le problème de la galerie d'art et la triangulation d'un polygone P
  - Ex. Polygone convexe : peut être gardé avec une seule caméra
  - Quel est le nombre de caméras nécessaires pour « garder » un polygone à n sommets
    - On ne cherche pas forcément le minimum ! (problème NP complet)
    - La réponse n'est à première vue pas évidente : un polygone à n cotés peut être de forme compliquée.
    - On va donc décomposer P en triangles, qui sont facile à « garder » individuellement puisque nécessairement convexes.

Triangulation de polygones

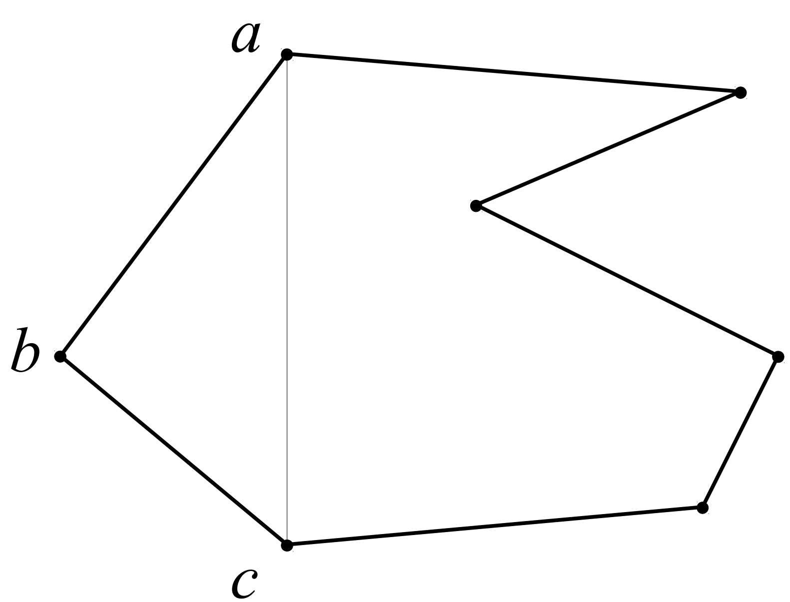
- On transforme en triangles en tirant des diagonales entre paires de sommets non reliés.



Existence de la triangulation d'un polygone

- Existence de la triangulation pour un polygone à n cotés: preuve par induction.

Pour n=3, c'est trivial. Posons $n > 3$ et supposons que la triangulation existe pour tout $m < n$. On va prouver l'existence d'une diagonale. Soit b le sommet le plus à gauche, et a et c les deux sommets immédiatement voisins ; Si le segment $\overline{ac}$ est compris dans l'intérieur du polygone, alors on a trouvé une diagonale.



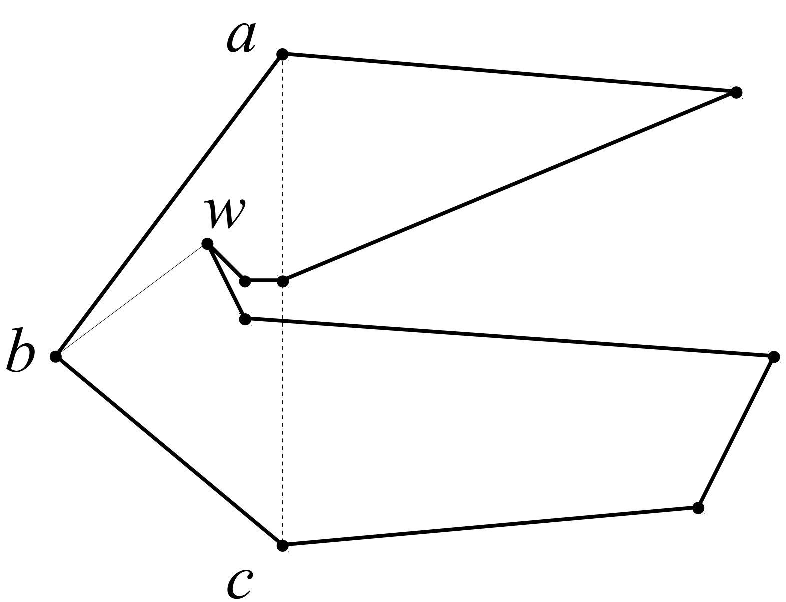
On décompose le polygone en deux, un triangle à p=3 sommets et un autre polygone à $q < n$ sommets, avec p+q=n+2. Dans le cas présent, q=n-1.

Existence de la triangulation d'un polygone

Si le segment $\overline{ac}$ n'est pas entièrement compris dans l'intérieur du polygone, alors prenons les points situés à l'intérieur du triangle abc.

Parmi ces points , soit w le plus éloigné de $\overline{ac}$.

Alors $\overline{bw}$ est une diagonale. Si ce n'était pas le cas ($\overline{bw}$ intersecte d'autres cotés du polygone) alors cela voudrait dire que w n'est pas le point le plus éloigné de $\overline{ac}$, ce qui est contradictoire.

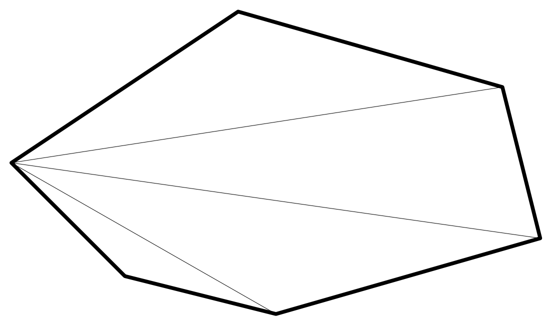


Dans ce cas , on décompose en deux polygones , un avec p sommets, l'autre avec q sommets, avec $p < n$ et $q < n$ et $p+q=n+2$.

- Dans les deux cas, on peut répéter l'opération autant de fois que nécessaire et arriver à une triangulation valide du polygone.

Existence de la triangulation d'un polygone

- Combien de triangles obtient-on ?



  - Pour un polygone convexe, on a $m=n-2$ triangles. Supposons que c'est vrai pour tout polygone, et en particulier pour les polygones avec $p,q < n$ sommets.

    En prenant une diagonale pour diviser en deux sous-polygones, dans les étapes précédentes : on a $p+q=n+2$, avec $p < n$, et $q < n$.

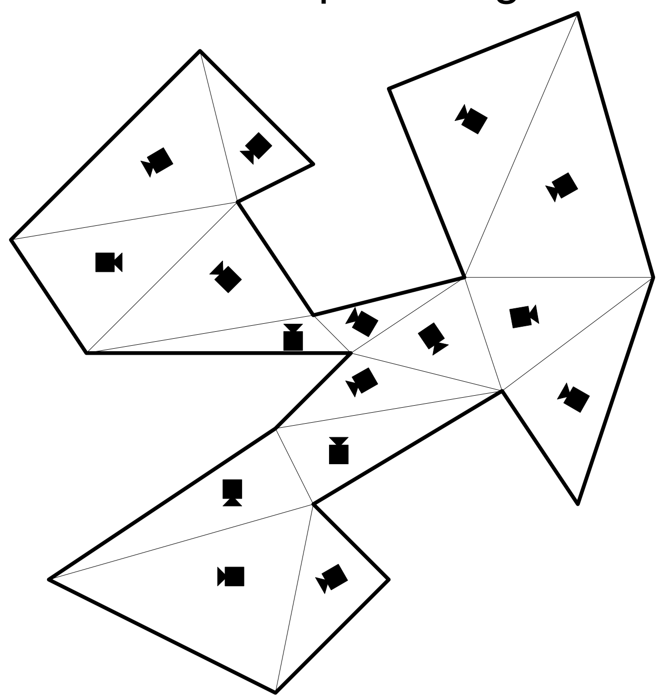
    Chaque sous-polygone se subdivise en $m_p=p-2$ et $m_q=q-2$.

    Donc le polygone initial est subdivisé en $m = m_p+m_q = p-2+q-2 = n+2-4 = n-2$ triangles. CQFD

Placement des caméras

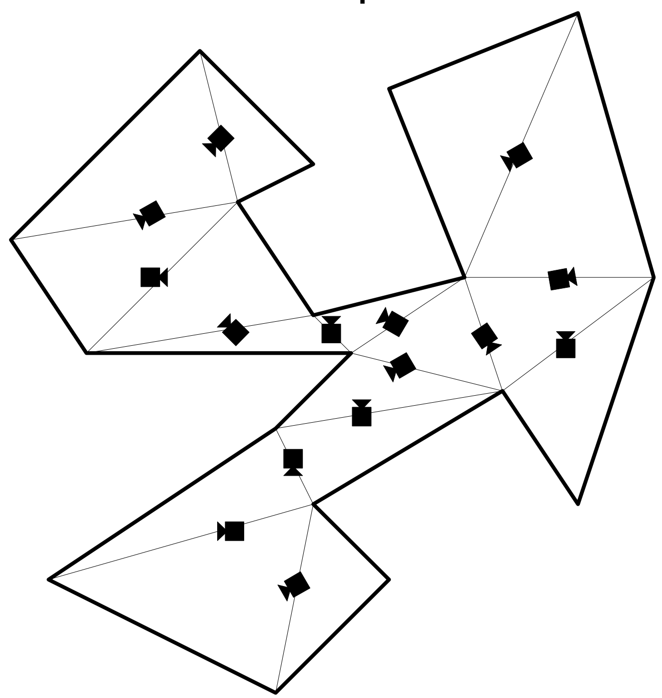
- Revenons au problème initial

1 caméra par triangle



$n-2$ caméras

1 caméra par arête

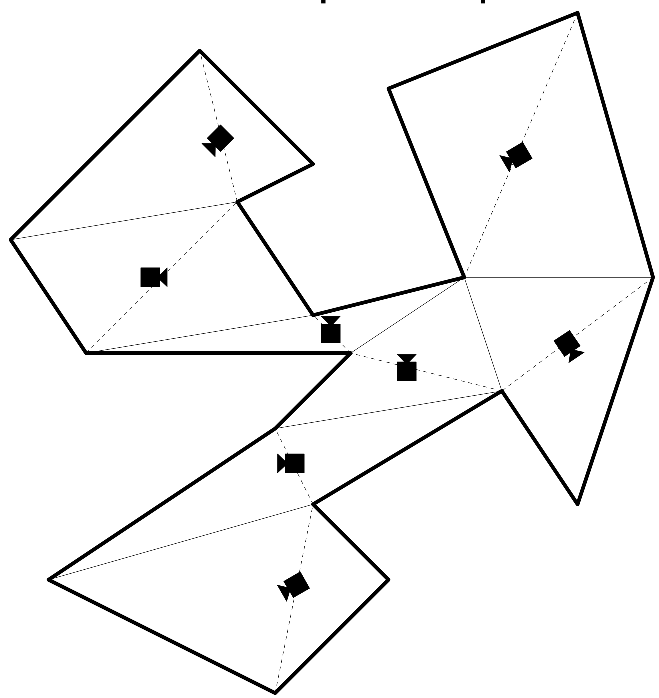


$n-3$ caméras

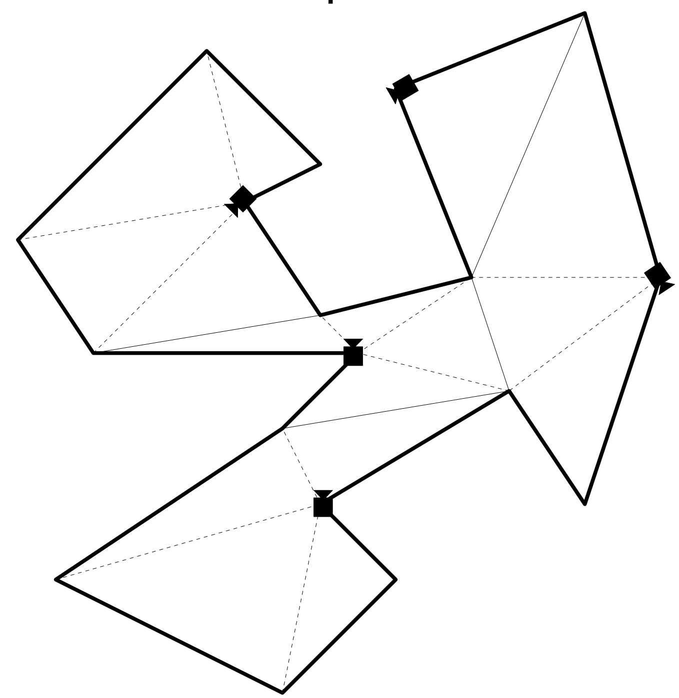
Placement des caméras

- Revenons au problème initial

1 caméra par couple de triangle



1 caméra pour certains sommets

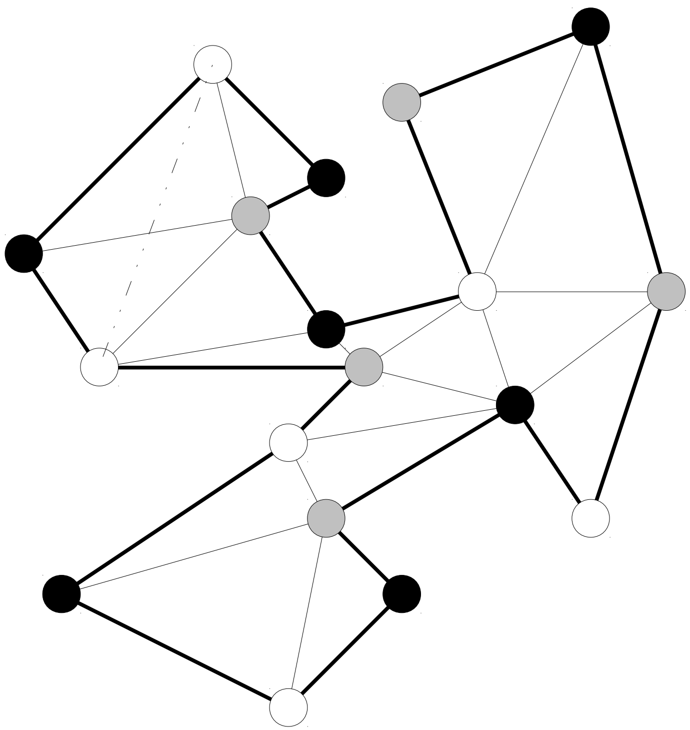


$\sim n/2$ caméras

$\sim n/3$ caméras ?

3-Coloriage

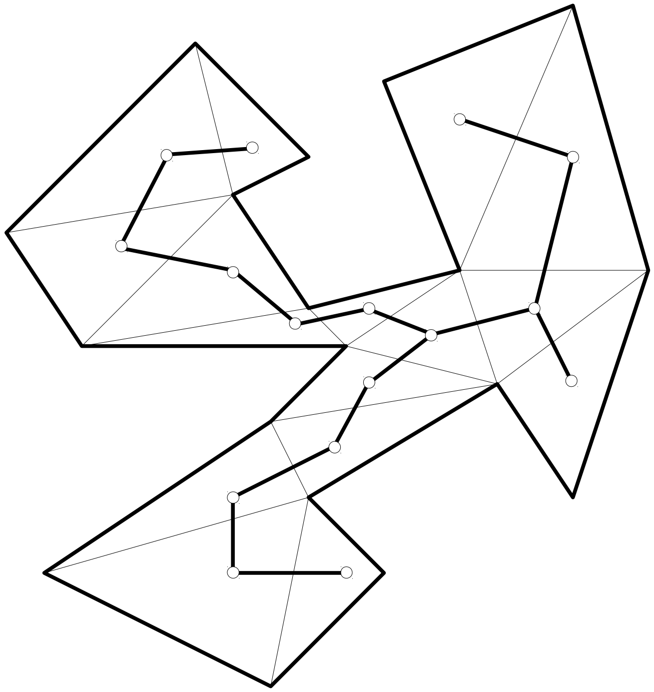
- Comment déterminer les sites où implanter une caméra ? → 3-coloriage.



- Chaque arête doit relier deux sommets de couleur différente
- On choisit de placer la caméra au sommets de la couleur la moins courante (ici, gris)
- Le 3-coloriage *dépend* de la triangulation choisie (si il est unique, ce n'est que pour une triangulation donnée)
- Existe-il toujours un tel coloriage ?

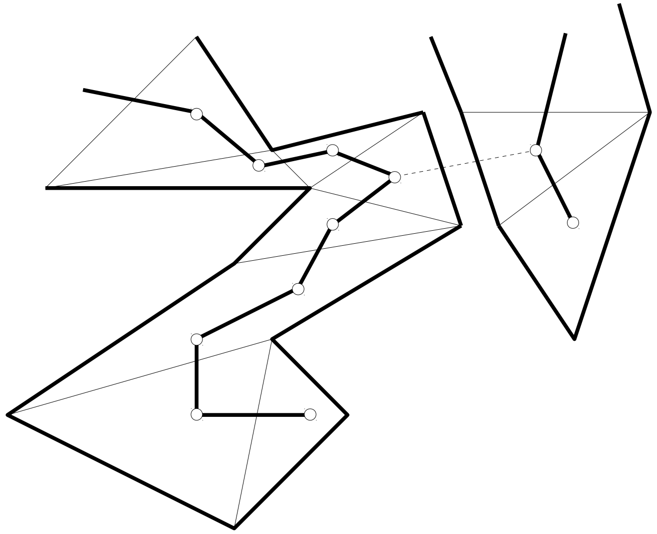
3-Coloriage

- Existence du 3-coloriage.



- Considérer le graphe de connectivité des triangles (graphe dual de la triangulation T)

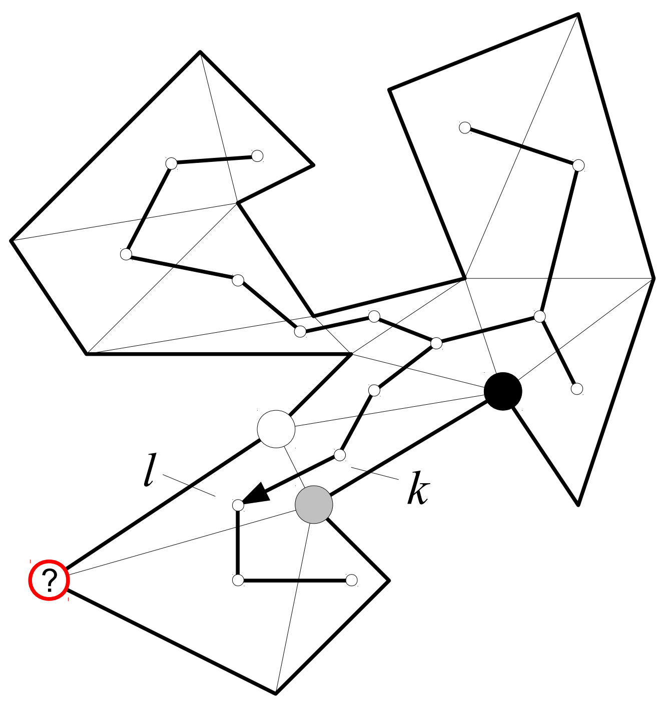
  Il s'agit nécessairement d'un arbre car si l'on retire une diagonale de T, on obtient deux triangulations, donc on coupe le graphe en deux parties disjointes.



Géométrie Algorithmique

3-Coloriage

- Existence du 3-coloriage.

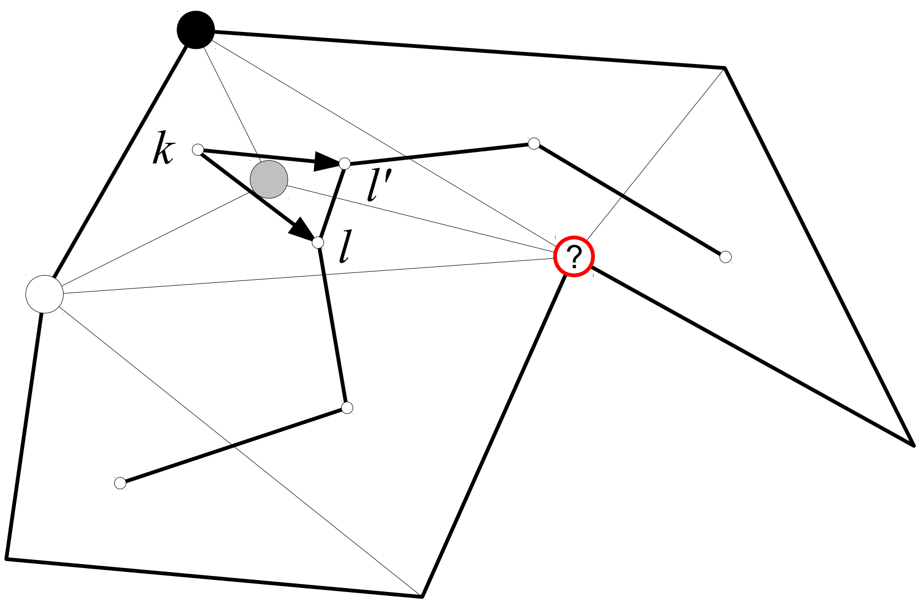


- On peut parcourir l'arbre en empruntant un segment en partant de n'importe quel noeud k. A chaque nouveau noeud, l par exemple, on sait que les deux triangles $t(k)$ et $t(l)$ partagent une arête, et celle ci est déjà colorié. Ne reste donc qu'un choix de couleur pour le sommet restant de $t(l)$.
- A chaque fois que l'on progresse dans l'arbre, on passe par une nouvelle arête de T ; par conséquent le choix de la couleur pour le sommet est toujours libre.
- Le coloriage peut être fait en $O(n)$ opérations.

3-Coloriage

- Notes :
  - Le 3-coloriage ne fonctionne que dans le cas ou le graphe est un arbre.
  - Dans le cas contraire, il est facile de trouver une contre exemple :

  - cycle à cause d'un sommet interne

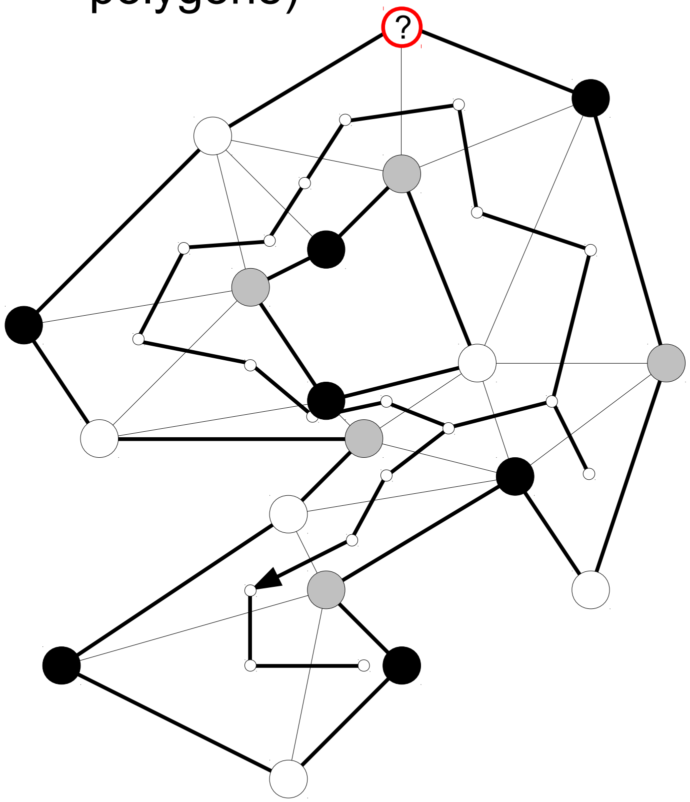


Il faut (au moins) une quatrième couleur

Géométrie Algorithmique

3-Coloriage

- Cycle dû à la présence de trous dans le patch (n'est plus un polygone)

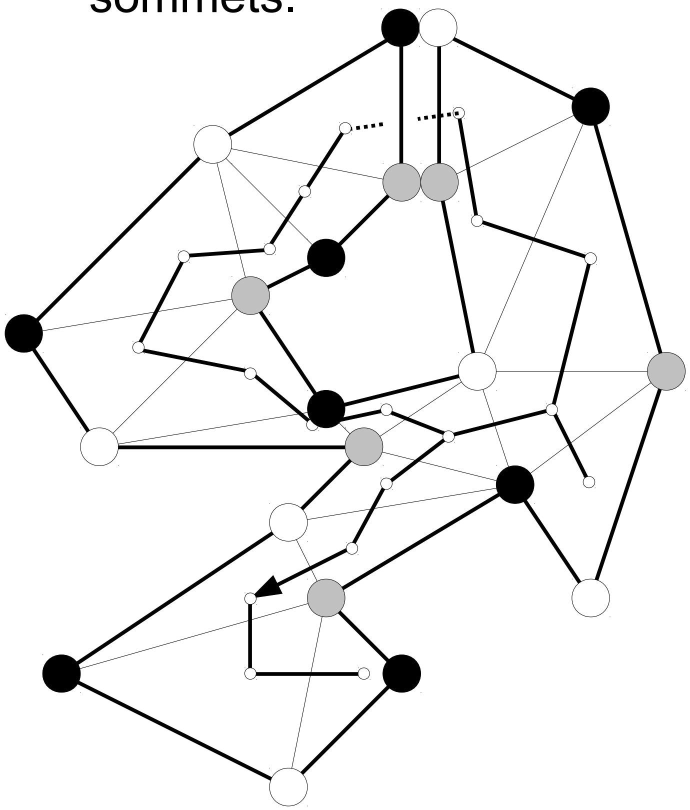


- Solution : transformer le patch en polygone en dédoublant certains sommets.

Géométrie Algorithmique

3-Coloriage

- Solution : transformer le patch en polygone en dédoublant certains sommets.



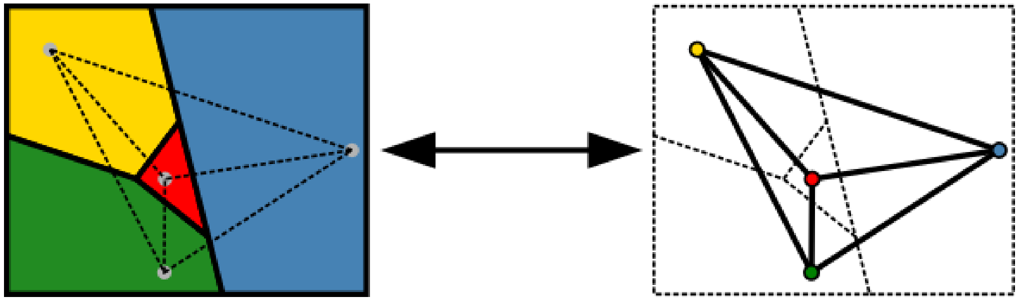
4-Coloriage

- En fait, le 4-coloriage existe toujours pour un graphe plan (donc à fortiori pour une triangulation)

  La conjecture date de 1852 , par un botaniste anglais qui note que le coloriage des comtés sur d'une carte d'Angleterre ne nécessite que 4 couleurs.

  - Démontré seulement en 1977 avec l'aide... d'ordinateurs.

  Appel, K. and Haken, W. "The Solution of the Four-Color Map Problem." Sci. Amer. 237, 108-121, 1977.



Placement des caméras

- Grâce au 3-coloriage, on peut noter que le nombre de caméras nécessaires est limité à la partie entière de $n/3$, dans le pire des cas.

  Il est bien entendu possible d'utiliser moins de caméras (polygone convexe p.ex. ,1 suffit)

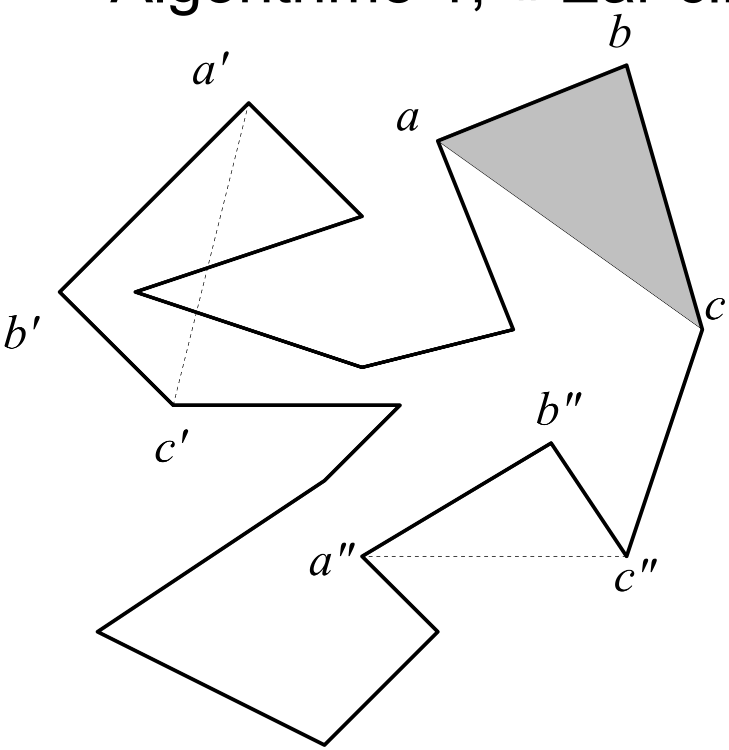
  On peut trouver des polygones pour lequel ce nombre est un minimum.

Exactement $n/3$ dents, exactement $n/3$ caméras.

- Revenons à nos moutons...

  Il nous faut générer une triangulation

  - Algorithme 1, « Ear clipping » (découpage des oreilles).



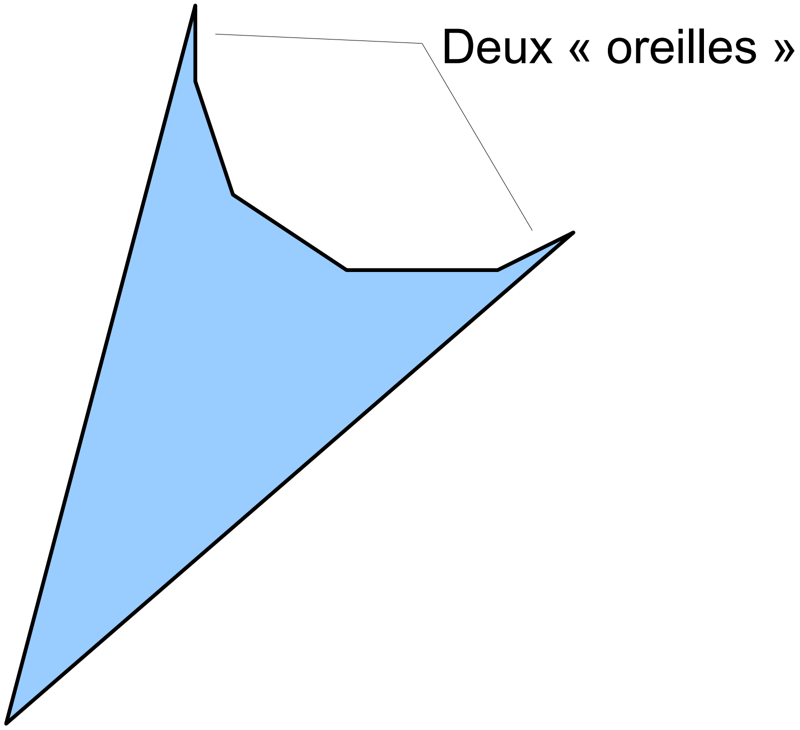
Une « oreille » est une suite a, b, c de trois sommets consécutifs formant un triangle situé dans le polygone, et qui ne contient pas d'autre sommets.

- Le principe de l'algorithme est de retirer successivement les « oreilles », au fur et à mesure...

  Mais un polygone a-t-il toujours des oreilles ? - OUI ! - il a au moins deux oreilles disjointes : cf

  Meisters, G. H., "Polygons have ears." American Mathematical Monthly 82 (1975). 648-651

Triangulation de polygones



Géométrie Algorithmique

Ear Clipping

Algorithme 1, « Ear clipping » (découpage des oreilles).

- Une implémentation naïve conduit à une complexité en $O(n^3)$.

  Trouver une oreille – $O(n^2)$

  L'éliminer – $O(1)$

  Trouver une nouvelle oreille – $O((n-i)^2)$

  Répété $n-2$ fois

- Il est possible de faire mieux : en $O(n^2)$

  Arranger les sommets dans 4 listes :
  - une contient tous les sommets (liste circulaire) – $O(n)$
  - une contient les sommets réflexes (concaves) – $O(n)$
  - une contient les sommets convexes – $O(n)$
  - enfin une contient les sommets des oreilles – $O(n^2)$

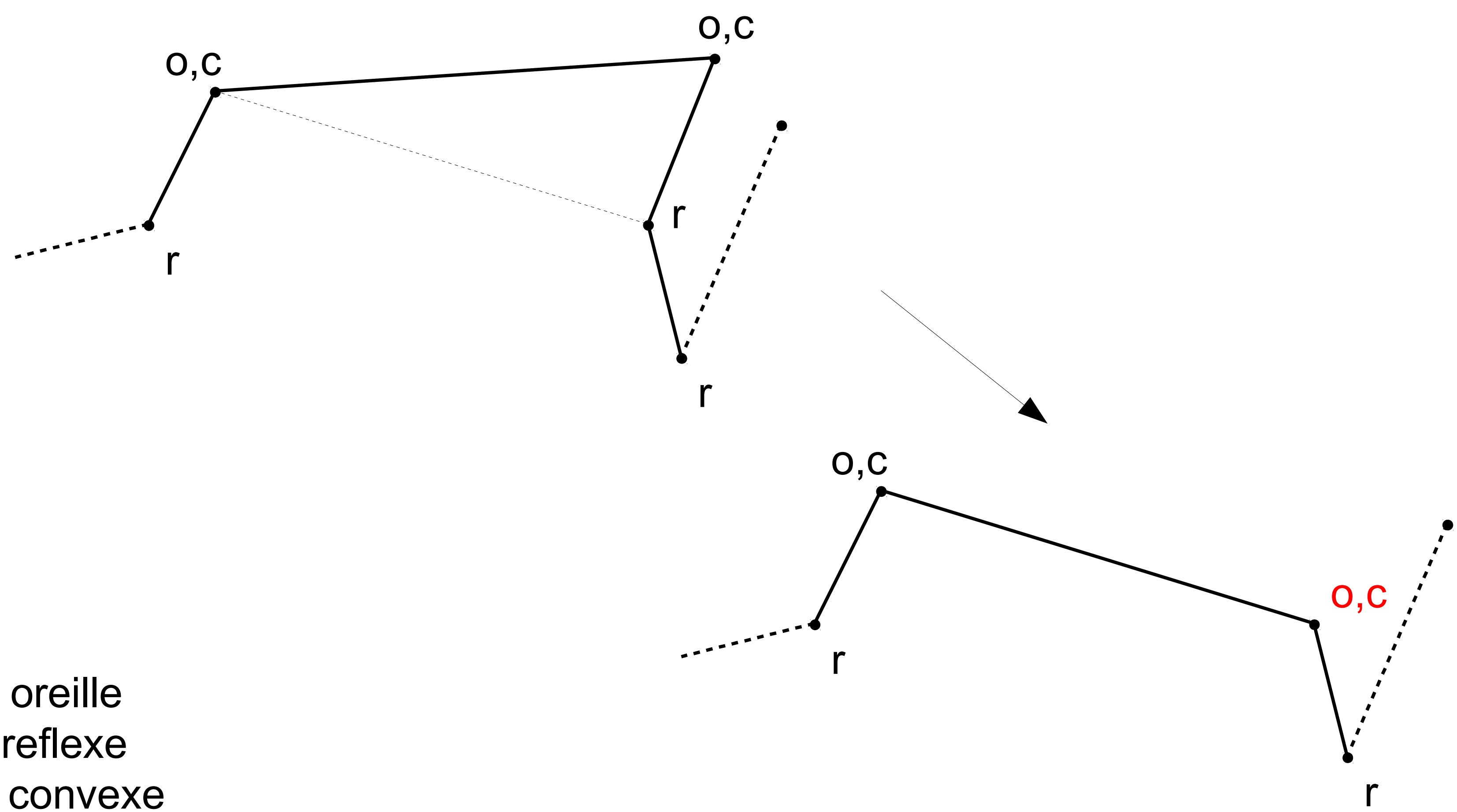
  Prendre la première oreille, l'éliminer – $O(1)$

  Vérifier les sommets immédiatement voisins dans les listes
  - Si il s'agissait d'un sommet convexe – il le reste
  - Si il s'agissait d'une oreille – il le reste
  - Si il s'agissait d'un sommet réflexe – il peut devenir convexe
  Il faut donc tester si il devient sommet d'une nouvelle oreille – en $O(n-i)$

  Mise à jour des listes – $O(1)$

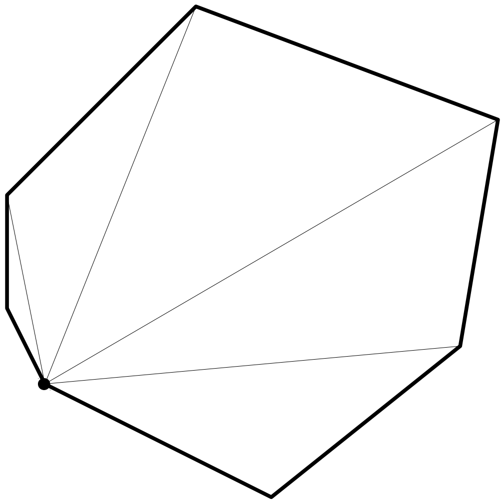
  $n-2$ fois

Ear Clipping

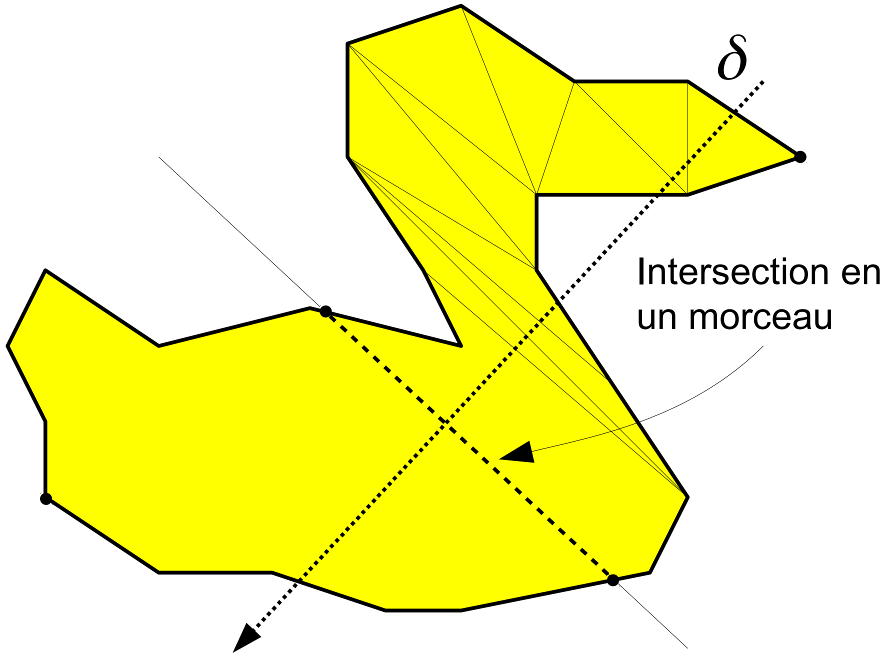


o = oreille
r = reflexe
c = convexe

- Peut-on faire mieux que $O(n^2)$?



Polygone convexe : $O(n)$



Polygone monotone
par rapport à la direction δ : $O(n)$
(démonstration formelle plus tard)

- Algorithme 2, Utilisation d'une décomposition en polygone plus simples à trianguler

  Idée : Décomposer le polygone de départ en un ensemble de sous-polygones pour lesquels la triangulation est plus simple.
  Mais : décomposition en polygone convexes aussi difficile que de faire la triangulation – reste les polygones monotones

  L'algorithme 2 est donc en deux parties :

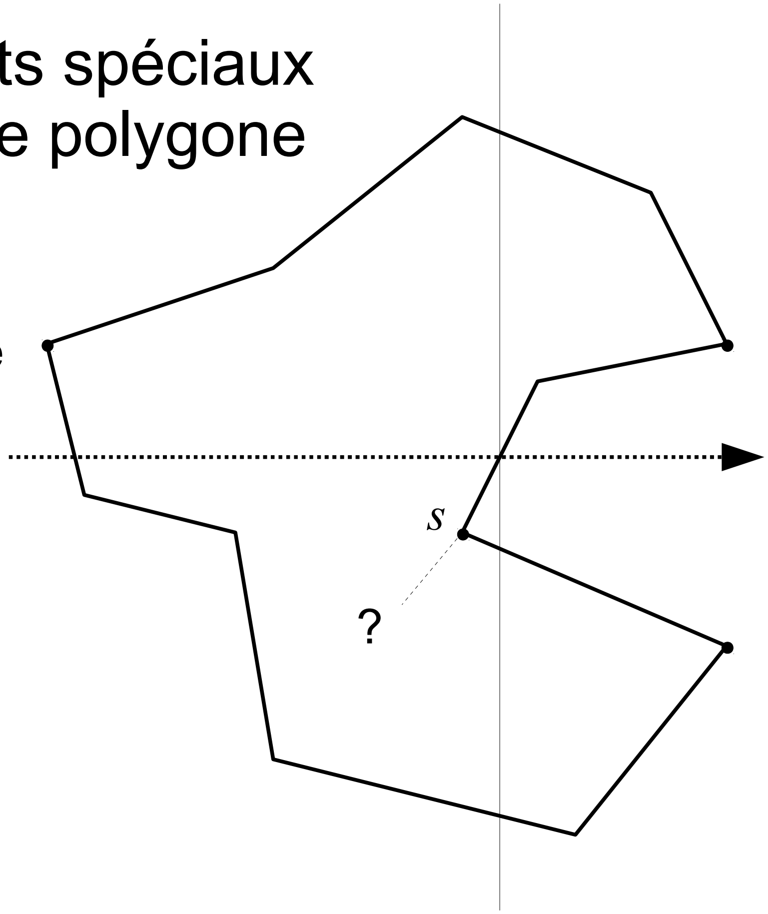
  – Décomposition du polygone en sous-polygones monotones selon une direction donnée (on espère cela en $o(n^2)$, soit p.ex. $O(n\log n)$)

  – Triangulation des sous-polygones

- Décomposition selon x

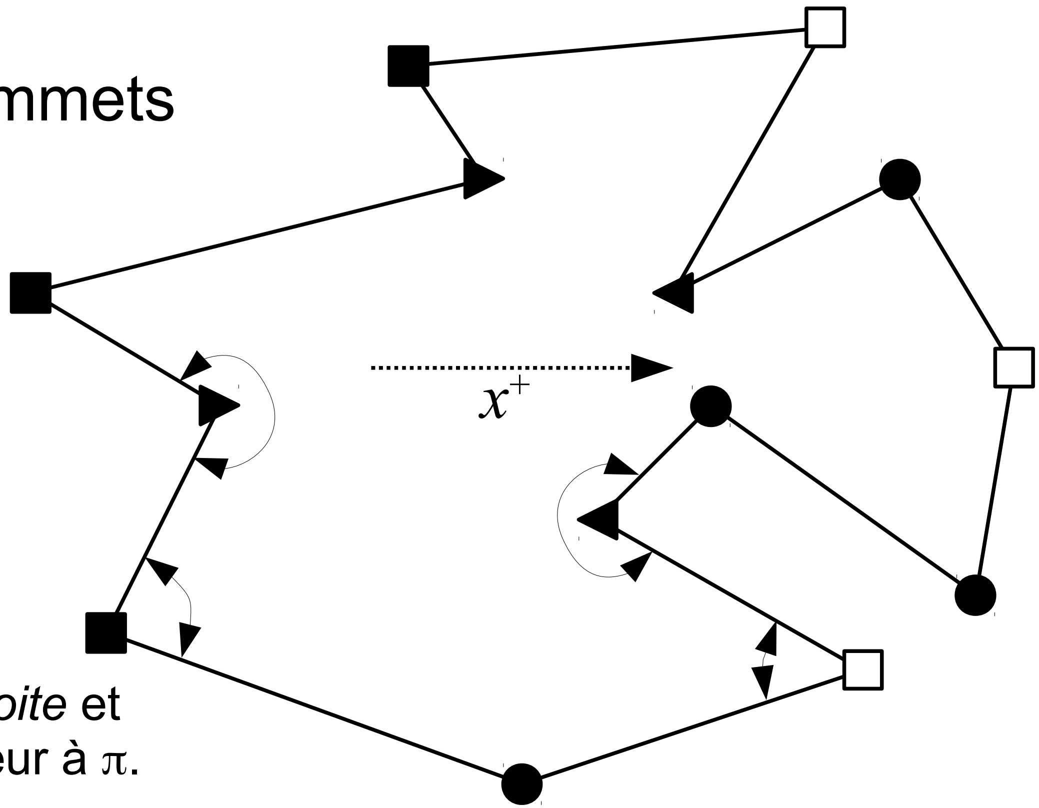
  Un certain nombre de sommets spéciaux existent. Ceux ci « rendent » le polygone non monotone.

  - Ces sommets sont les sommets pour lesquels il y a une inversion de sens de parcours apparent (par rapport à la direction de référence) lorsque l'on suit l'arc du polygone.
  - C'est à partir de ces sommets que l'on va ajouter des diagonales pour construire la décomposition



Décomposition en polygones monotones

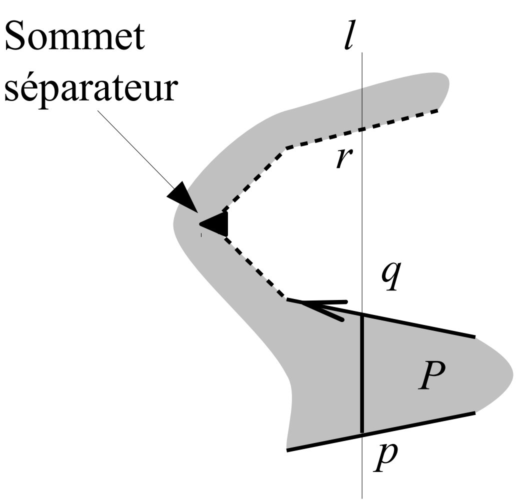
- Classification des sommets
  - ● Sommet « régulier »
  - ■ Sommet « début »
  - □ Sommet « fin »
  - ◀ Sommet « séparateur »
  - ▶ Sommet « fusion »



- Le sommet « **début** » a deux voisins situés *à droite* et un angle intérieur inférieur à π.
  - Si l'angle est supérieur à π le sommet est un sommet « **séparateur** »
- Le sommet « **fin** » a deux voisins à gauche et un angle intérieur inférieur à π
  - Si l'angle est supérieur à π le sommet est un sommet « **fusion** »
- Pour tous les autres cas, le sommet est « **régulier** »

Décomposition en polygones monotones

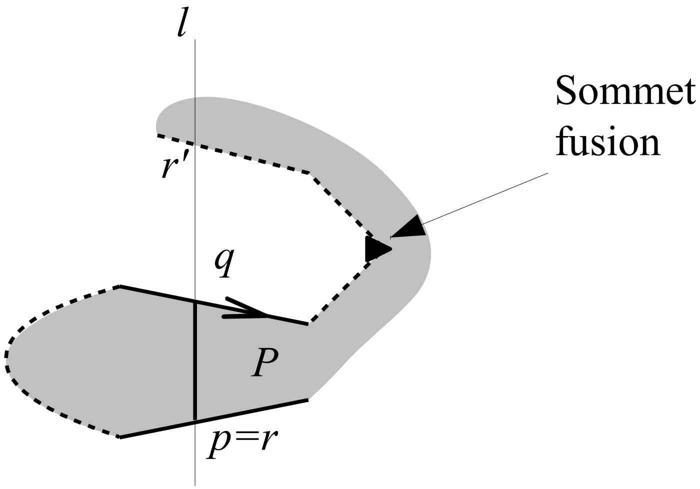
- Il est évident qu'un polygone est monotone (ici, selon x) si il ne contient pas de sommet « fusion » ni de sommet « séparateur »
  - Revient à prouver que si un polygone n'est pas monotone, alors il contient au moins un sommet « fusion » ou « séparateur »
  - Non x-monotone → une ligne verticale l intersecte P en plus d'une composante connexe. Choisissions l telle que la plus basse soit un segment – pas un simple sommet.



- Soit p le point sous la frontière, et q le point supérieur.
- Partons de q et allons dans la direction telle que le polygone se trouve à la gauche de la frontière.
- À un moment donnée, on intersecte l à nouveau, en r.
- Si $r \neq p$, alors on a rencontré un point extrême qui est un nécessairement un séparateur.
- Si $r = p$, on va parcourir la frontière dans l'autre sens.

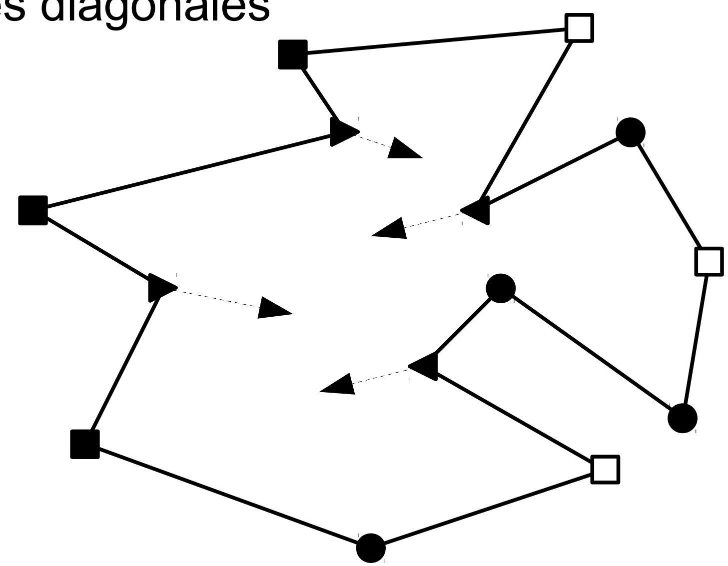
Décomposition en polygones monotones

- Si $r = p$, on va parcourir la frontière dans l'autre sens. Comme précédemment, on intersecte l en un nouveau point r'. On a nécessairement $r' \neq p$, sinon cela voudrait dire que l'intersection de l avec P ne comporte qu'un seul segment connexe.
- Alors il existe un sommet le plus à droite, qui est nécessairement un sommet séparateur.
- Donc un polygone non x-monotone possède nécessairement un sommet séparateur , et/ou un sommet fusion



Décomposition en polygones monotones

- Cela implique que si l'on parvient à éliminer (*i.e.* transformer en autre types de sommets) les sommets « séparateurs » et « fusion », alors on aura généré des polygones monotones.
  - Ceci est effectué en ajoutant des diagonales : une diagonale vers la gauche pour chaque sommet « séparateur », et une diagonale vers la droite pour chaque sommet « fusion ».
  - La difficulté consiste à « relier » les diagonales au reste du polygone.



Décomposition en polygones monotones

- Utilisation du balayage plan (encore !)
  - Les événements sont les sommets du polygone (pas de nouvel événement créé au fur et a mesure)
  - Ils sont traités par ordre lexicographique (classés par x puis par y si x égal), dans une file de priorité.
  - Le statut T permet de construire les diagonales au fur et à mesure du traitement des événements

Décomposition en polygones monotones

- Que faire quand l rencontre un événement ?
  - Cas des sommets « séparateurs »
    - Il faut relier celui ci à un sommet proche, de telle façon que l'on intersecte pas d'autres arêtes

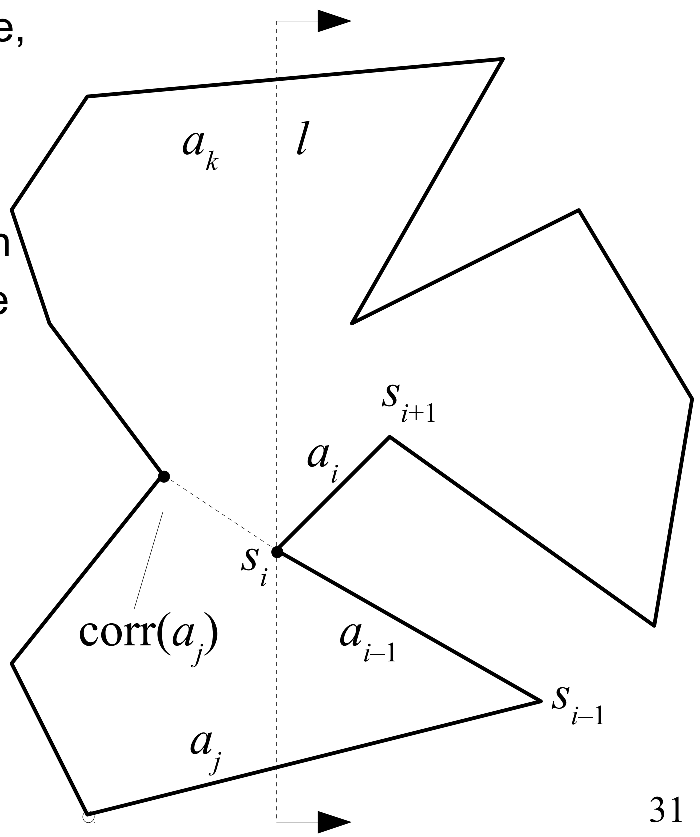
      Soit a_j l'arête immédiatement sous s_i le long de l, et a_k celle au dessus. Alors on peut toujours connecter s_i au sommet le plus à droite situé entre a_j et a_k, et à la gauche de s_i. Si il n'existe pas de tel sommet, on peut utiliser le sommet le plus à gauche de a_j (ou de a_k).

      Dans tous les cas, ce sommet est nommé correspondant de a_j : $\mathrm{corr}(a_j)$.

s_i dans sens antihoraire

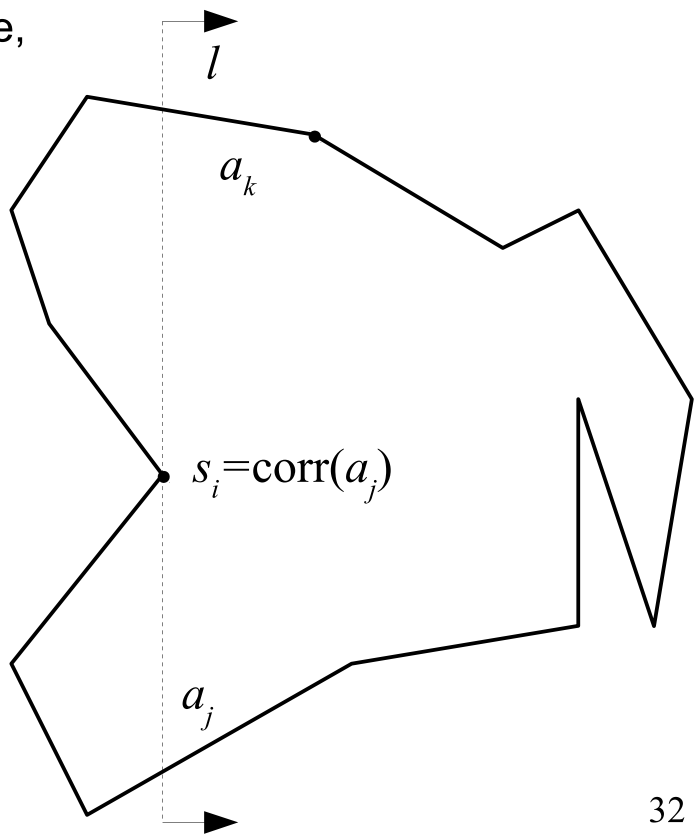
$a_i = \overline{s_i s_{i+1}}$

$a_{n-1} = \overline{s_{n-1} s_0}$



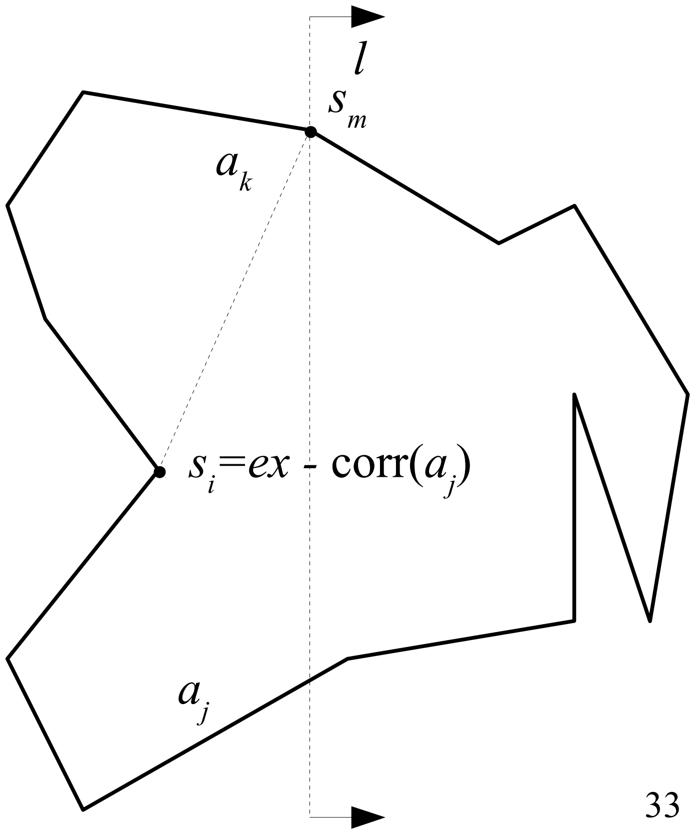
Décomposition en polygones monotones

- Que faire quand l rencontre un événement ?
  - Cas des sommets « Fusion »
    - Il faut relier celui ci à un sommet proche, de telle façon que l'on intersecte pas d'autres arêtes – MAIS on ne peut pas le faire immédiatement (le sommet potentiel étant situé *après* l)
    - On peut observer que s_i peut être considéré comme le correspondant de a_j .



Décomposition en polygones monotones

- Que faire quand l rencontre un événement ?
  - Cas des sommets « Fusion »
    - Lorsque l progresse, on rencontre s_m qui devient le nouveau correspondant de a_j .
    - C'est à ce moment que l'on peut relier s_m à s_i .
    - Ceci n'est fait que si s_i est un sommet fusion !
    - Il se peut que s_m lui même soit un sommet séparateur – tant mieux ! (une diagonale pour deux)



Décomposition en polygones monotones

- Statut à mettre à jour à chaque événement :
  - Arbre de recherche binaire T – contient les arêtes pour lesquelles l'intérieur du polygone est au dessus.
  - Le classement dans cet arbre se fait en fonction de la position verticale. Les arêtes les plus basses sont en premier.
  - Pour chaque arête stockée dans l'arbre, on stocke le « correspondant »
- Chaque événement s_i on met à jour ce statut.
  - s_i est un sommet « début » : insère l'arête a_i dans T, met à jour $\mathrm{corr}(a_i)=s_i$.
  - s_i est un sommet « fin » : si $\mathrm{corr}(a_{i-1})$ est un sommet « fusion » alors crée une diagonale entre s_i et $\mathrm{corr}(a_{i-1})$. Efface a_{i-1} de T.
  - s_i est un sommet «séparateur »
    Recherche dans T l'arête a_j située sous s_i
    Crée une diagonale entre s_i et $\mathrm{corr}(a_j)$
    Met à jour $\mathrm{corr}(a_j)=s_i$
    Insère a_i dans T et pose $\mathrm{corr}(a_i)=s_i$.

Décomposition en polygones monotones

- s_i est un sommet « fusion »

  si $\mathrm{corr}(a_{i\text{-}1})$ est un sommet fusion

  créer une diagonale entre s_i et $\mathrm{corr}(a_{i\text{-}1})$

  retire $a_{i\text{-}1}$ de T

  recherche dans T l'arête a_j directement sous s_i

  si $\mathrm{corr}(a_j)$ est une sommet fusion

  crée une diagonale entre s_i et $\mathrm{corr}(a_j)$

  met à jour $\mathrm{corr}(a_j)=s_i$.

- s_i est un sommet « régulier »

  Si l'intérieur de P est « au dessus » de s_i

  Si $\mathrm{corr}(a_{i\text{-}1})$ est est un sommet fusion

  crée la diagonale entre s_i et $\mathrm{corr}(a_{i\text{-}1})$

  efface $a_{i\text{-}1}$ de T

  insère a_i dans T et met à jour $\mathrm{corr}(a_i)=s_i$

  sinon recherche dans T de a_j qui est en dessous de s_i

  si $\mathrm{corr}(a_j)$ est un sommet « fusion »

  crée une diagonale entre s_i et $\mathrm{corr}(a_j)$

  met à jour $\mathrm{corr}(a_j) = s_i$

- Performances :
  - Construire la file d'événements prend $O(n \log n)$
  - Chaque événement, on prend un temps $O(\log n)$ (au pire) et il y a n événements.
  - Le stockage requis est en $O(n)$
  - En tout, $O(n \log n)$

Triangulation de polygones monotones

- Traitement des sommets dans l'ordre lexicographique
- Utilisation d'une pile P
  - Contient des sommets que l'on a rencontrés mais qui doivent encore être reliés à d'autres – ceux ci sont sous forme d'un cône.
  - Deux cas :
    - le prochain sommet traité est du coté du 1er sommet de la pile
    - Ou non...

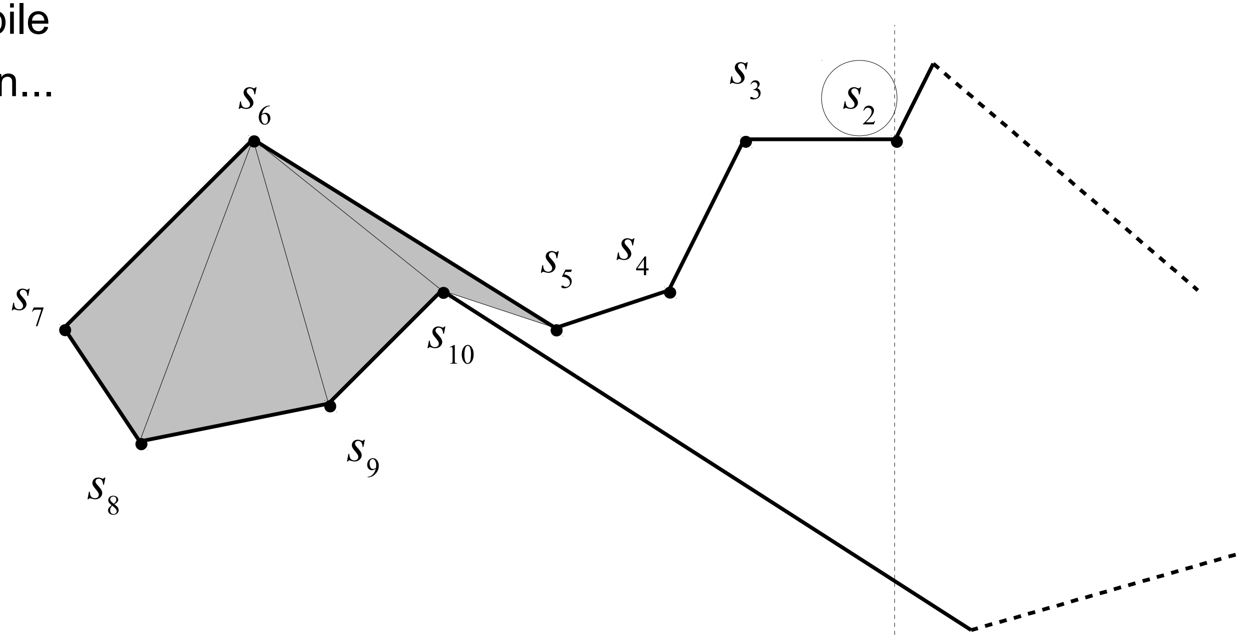
Pile P

s_3

s_4

s_5

s_{10}



Géométrie Algorithmique

Triangulation de polygones monotones

- Cas ou le prochain sommet traité EST du coté du 1er sommet de la pile
  - On peut tenter de relier successivement les sommets de la pile au sommet courant.
  - A un moment, on doit s'arrêter (une diagonale coupe le polygone)

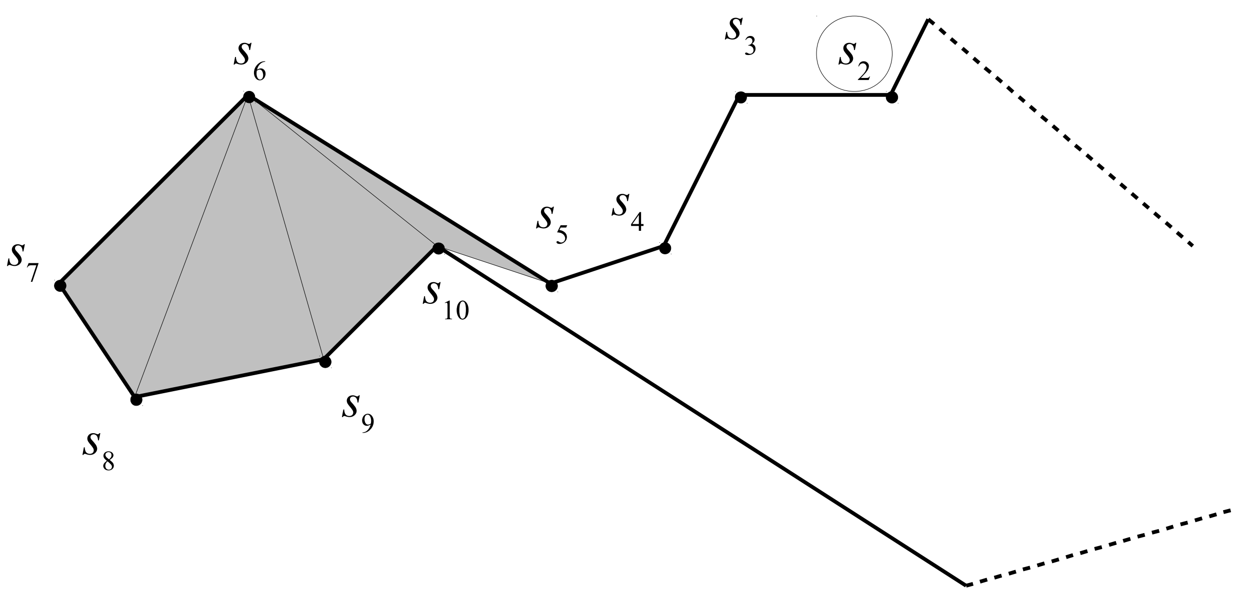
Pile P

s_3

s_4

s_5

s_{10}



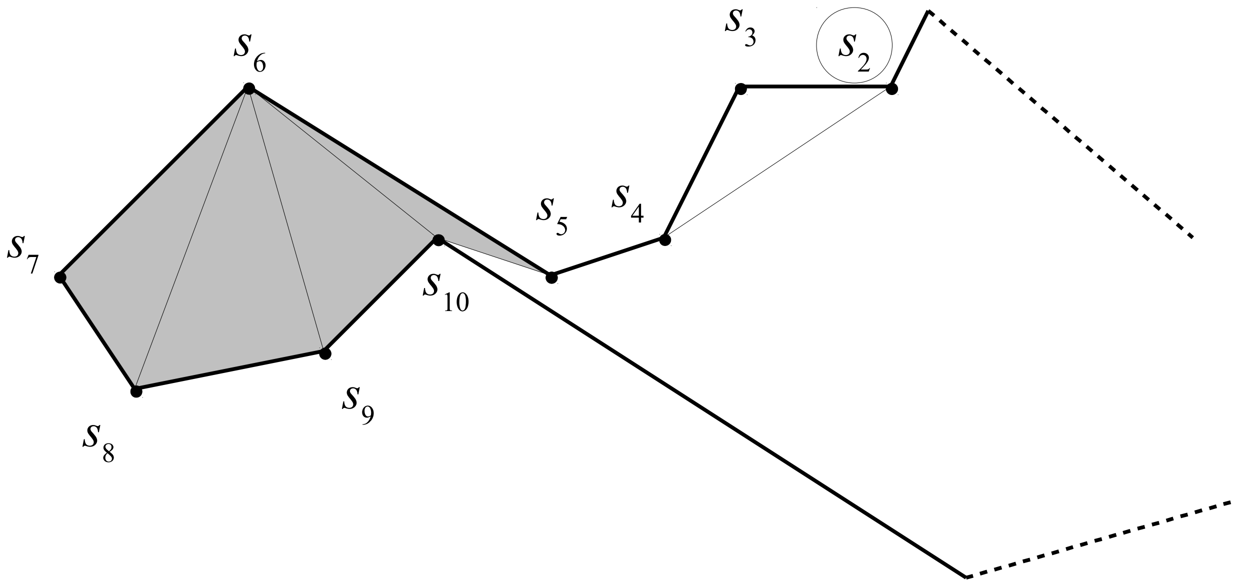
Triangulation de polygones monotones

- Cas ou le prochain sommet traité EST du coté du 1er sommet de la pile
  - On peut tenter de relier successivement les sommets de la pile au sommet courant.
  - A un moment, on doit s'arrêter (une diagonale coupe le polygone)

Pile P

s_5

s_{10}



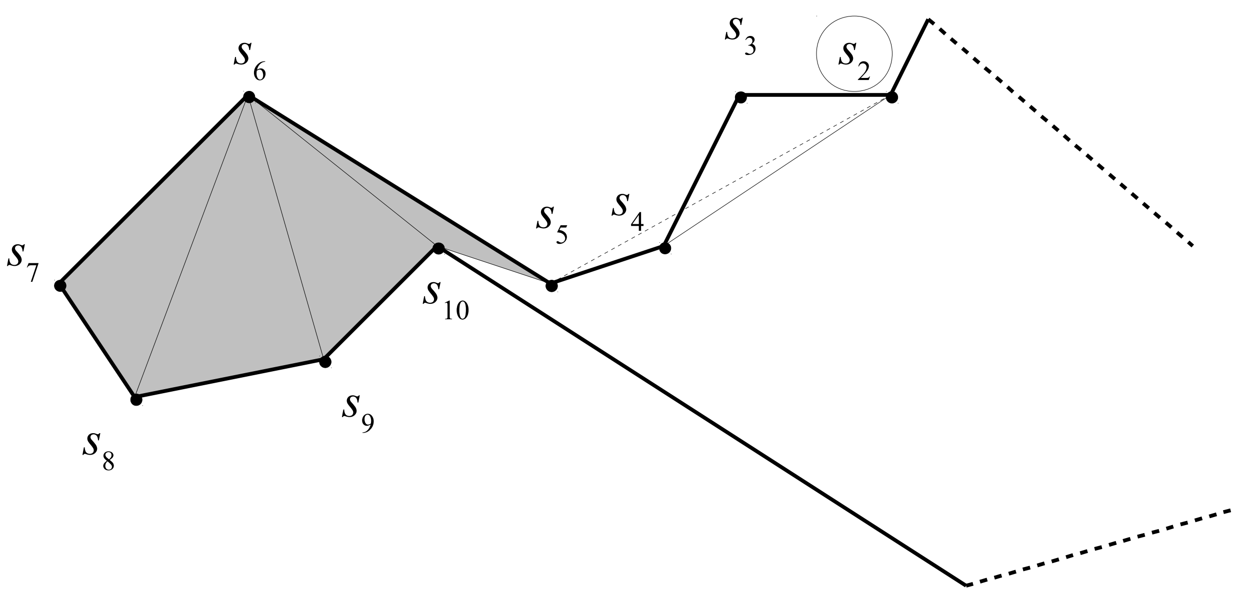
Triangulation de polygones monotones

- Cas ou le prochain sommet traité EST du coté du 1er sommet de la pile
  - On peut tenter de relier successivement les sommets de la pile au sommet courant.
  - A un moment, on doit s'arrêter (une diagonale coupe le polygone)

Pile P

s_5

s_{10}



Triangulation de polygones monotones

- Cas ou le prochain sommet traité EST du coté du 1er sommet de la pile
  - On peut tenter de relier successivement les sommets de la pile au sommet courant.
  - A un moment, on doit s'arrêter (les diagonales coupent le polygone)

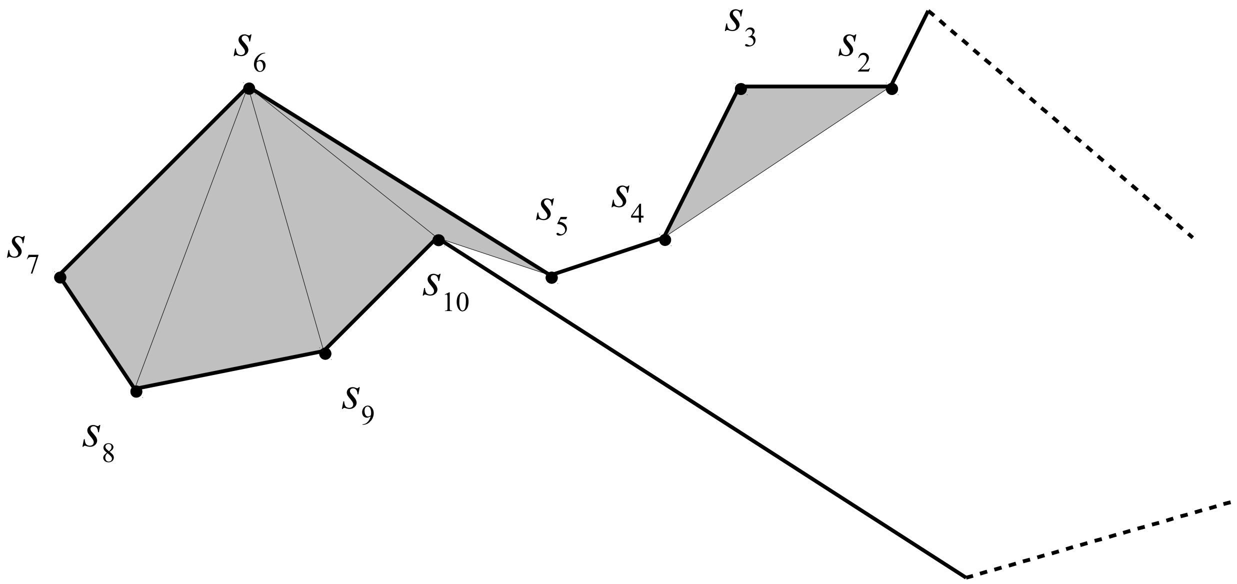
Pile P

s_2

s_4

s_5

s_{10}



  - En fin de procédure, on ré-empile le premier et le dernier sommet reliés

Triangulation de polygones monotones

- Cas ou le prochain sommet traité EST du coté du 1er sommet de la pile
  - On peut tenter de relier successivement les sommets de la pile au sommet courant.
  - Il se peut que l'on ne fasse qu'empiler le sommet courant ...

Pile P

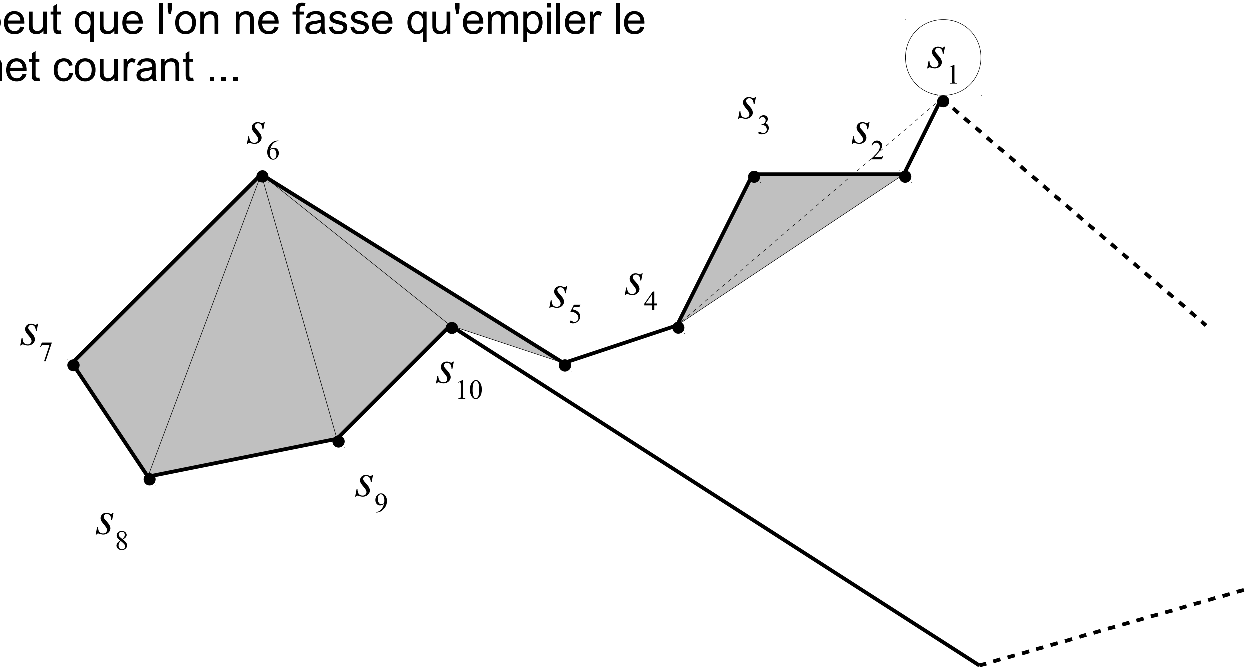
s_1

s_2

s_4

s_5

s_{10}



Triangulation de polygones monotones

- Cas où le prochain sommet traité N'EST PAS du coté du 1er sommet de la pile

Pile P

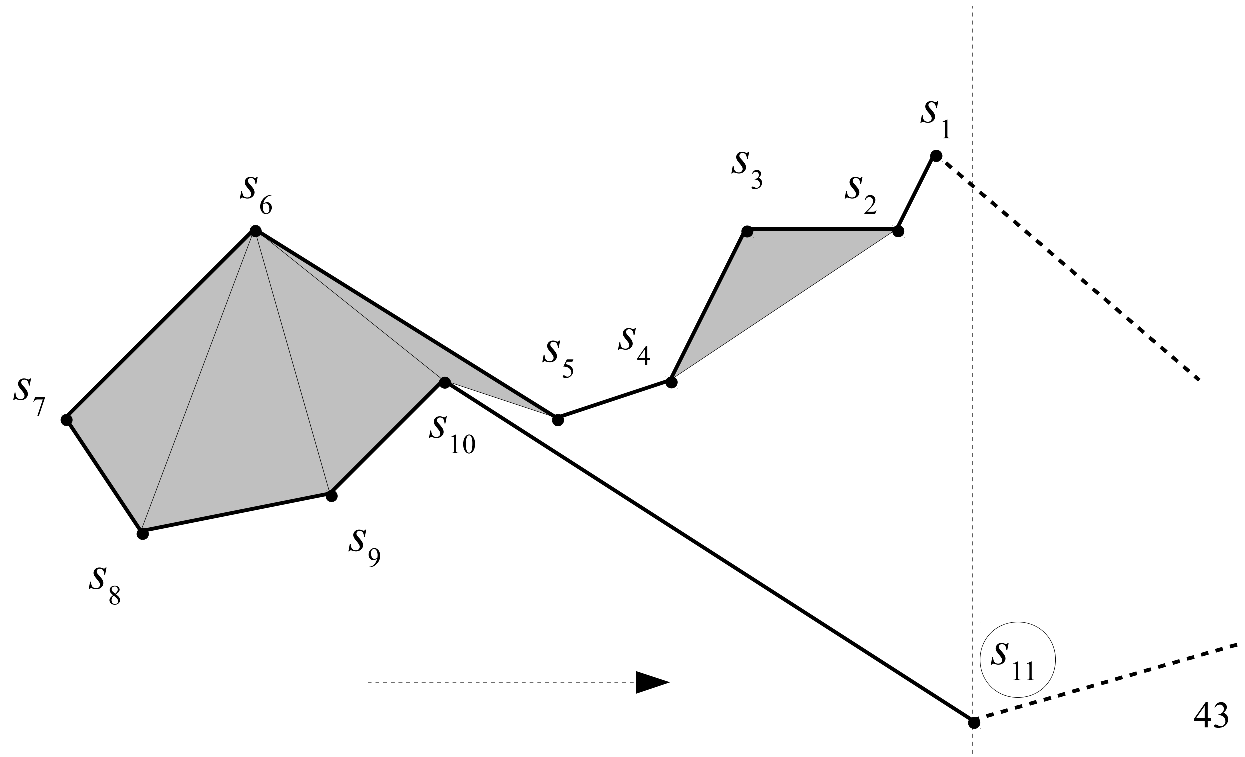
s_1

s_2

s_4

s_5

s_{10}



Triangulation de polygones monotones

- Cas ou le prochain sommet traité N'EST PAS du coté du 1er sommet de la pile
  - On peut relier successivement tous les sommets de la pile (sauf le dernier) au sommet courant

Pile P

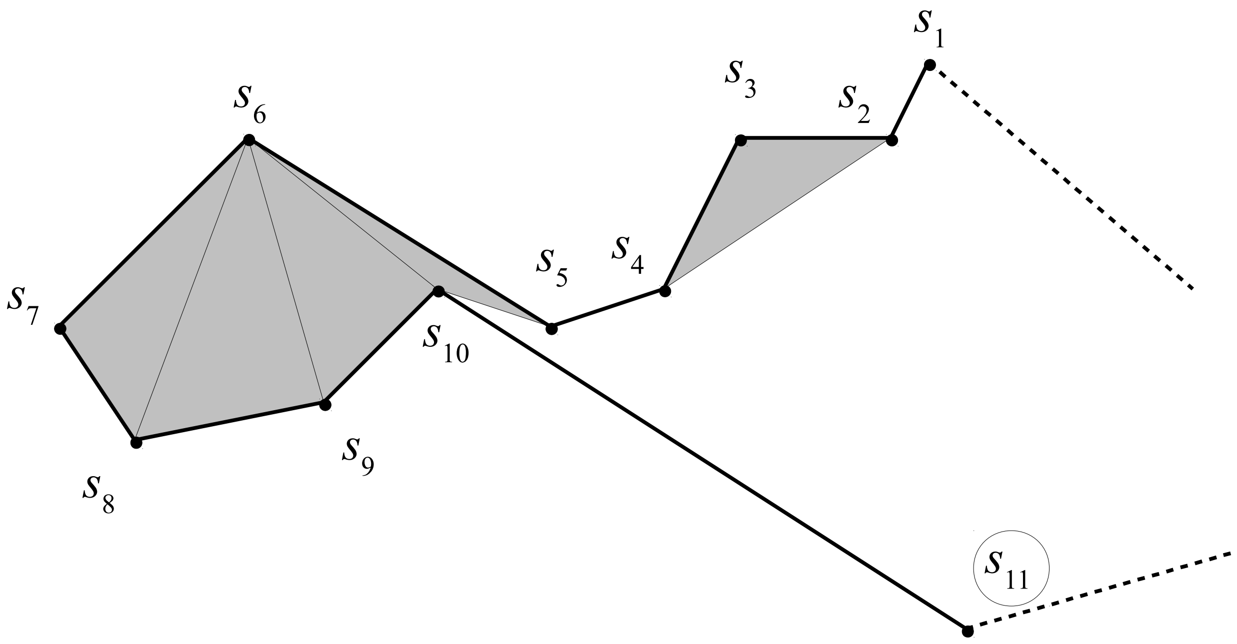
s_1

s_2

s_4

s_5

s_{10}



Géométrie Algorithmique

Triangulation de polygones monotones

- Cas ou le prochain sommet traité N'EST PAS du coté du 1er sommet de la pile
  - On peut relier successivement tous les sommets de la pile (sauf le dernier) au 1er sommet.

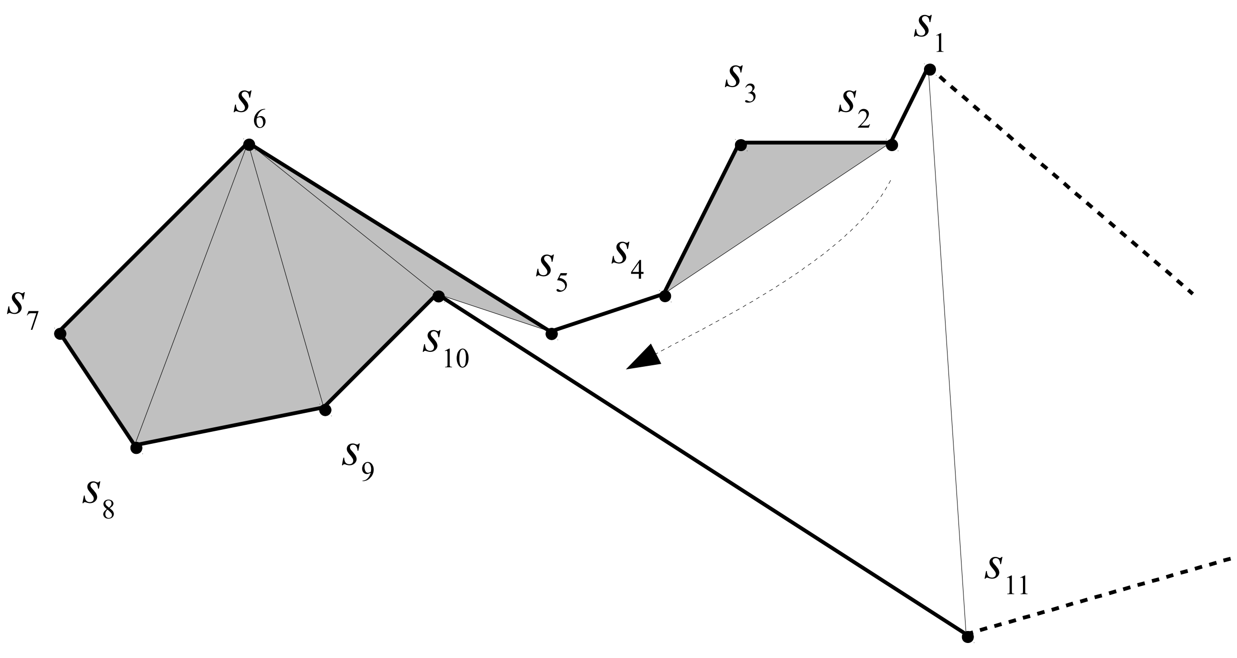
Pile P

s_2

s_4

s_5

s_{10}



Triangulation de polygones monotones

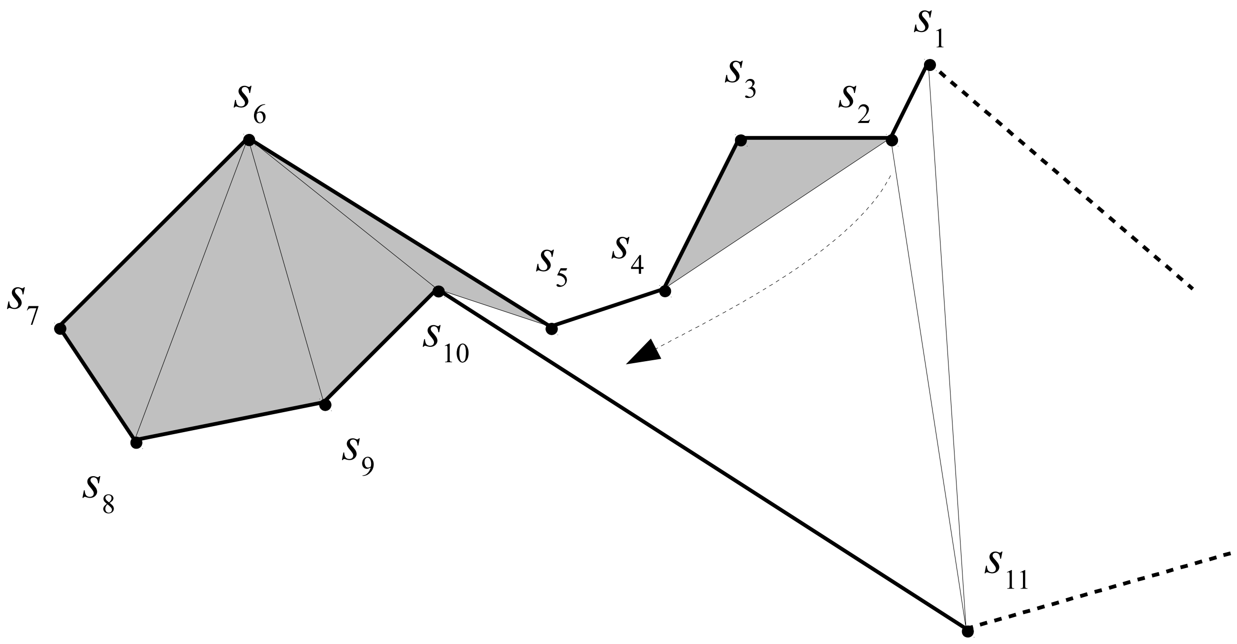
- Cas ou le prochain sommet traité N’EST PAS du coté du 1er sommet de la pile
  - On peut relier successivement tous les sommets de la pile (sauf le dernier) au 1er sommet.

Pile P

s_4

s_5

s_{10}



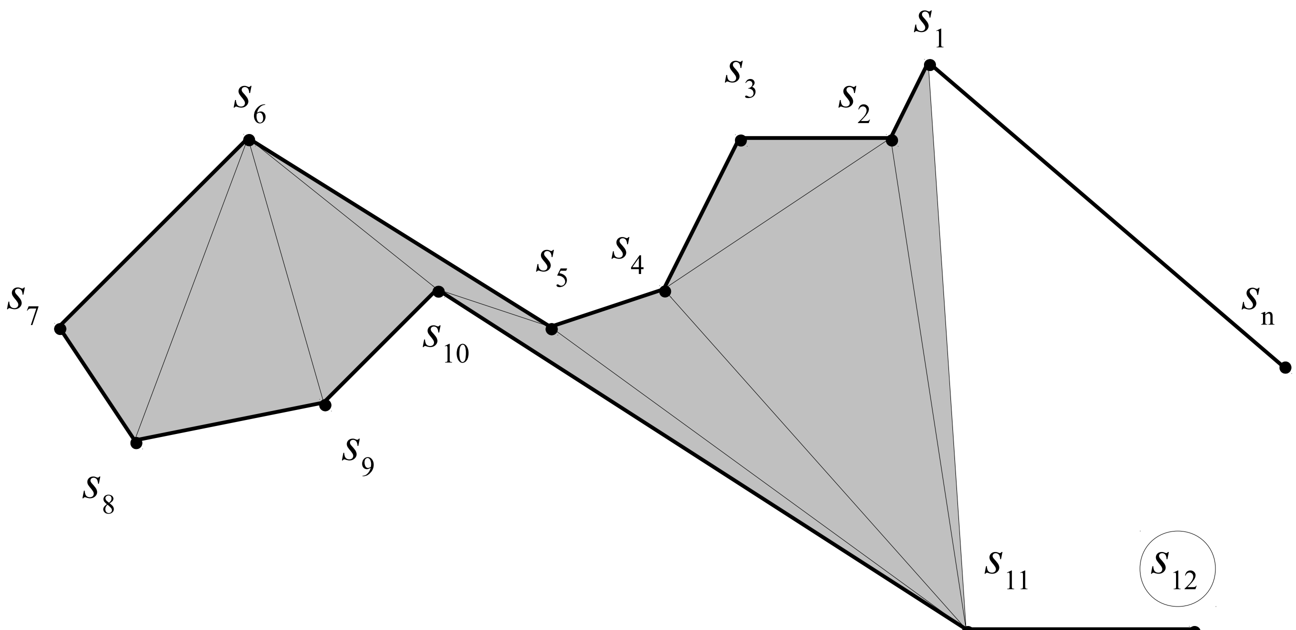
Triangulation de polygones monotones

- Cas ou le prochain sommet traité N'EST PAS du coté du 1er sommet de la pile
  - On peut relier successivement tous les sommets de la pile (sauf le dernier) au 1er sommet.

Pile P

s_{11}

s_1



- En fin de procédure, on ré-empile l'ex premier sommet de la pile et le sommet courant.
- On traite le sommet suivant : par exemple s_{12}

Triangulation de polygones monotones

TriangulationPolygoneMonotone(PM)
En entrée : un polygone x-monotone PM décrit p.ex. dans un DCEL
En sortie : une triangulation de PM
{
Fusionner les sommets dans la chaine supérieure et inférieur dans une structure commune. Les sommets sont classés selon un ordre lexicographique , t_0 est le premier sommet (plus à gauche), t_{n-1} est le dernier.
Initialiser une pile P vide, et y empiler t_0 et t_1
Pour j de 2 à n-2
{
Si t_j et le premier sommet de la pile P sont sur des chaines différentes
{
Dépiler tous les sommets de la pile, insérer une diagonale entre ceux ci et le sommet courant tj, sauf le dernier.
Empiler t_{j-1} et t_j
}
Sinon
{
Dépiler un sommet de P ;
Dépiler un à un les autres sommets de P tant que l'on peut tirer une diagonale
Empiler le dernier sommet dépilé, et t_j
}
}
Ajouter des diagonales entre t_{n-1} et tous les sommets restants dans la pile, sauf le premier et le dernier
}

Triangulation de polygones monotones

- Complexité de l'algorithme de triangulation de polygones monotones
  - Le classement des sommets est en $O(n)$ car le polygone est monotone !
  - La boucle est effectuée n-3 fois
    - Donc, on a au pire $2n$-4 empilages
    - Le nombre de dépilages ne peut excéder celui d'empilages, donc est globalement borné par $2n$-4
    - Chaque opération individuelle peut être faite en temps constant $O(1)$
- En conséquence, la complexité de l'algorithme est $O(n)$

Triangulation de polygones

- Complexité de l'algorithme de triangulation au complet
  - La décomposition est en $O(nlogn)$
  - Elle génère k polygone monotones , chacun de ceux ci peut être triangulé en $O(n_i)$. On a $\Sigma n_i = O(n)$
    - Donc la triangulation est en $O(n)$
- En conséquence, la complexité de l'ensemble de l'algorithme est $O(nlogn)$

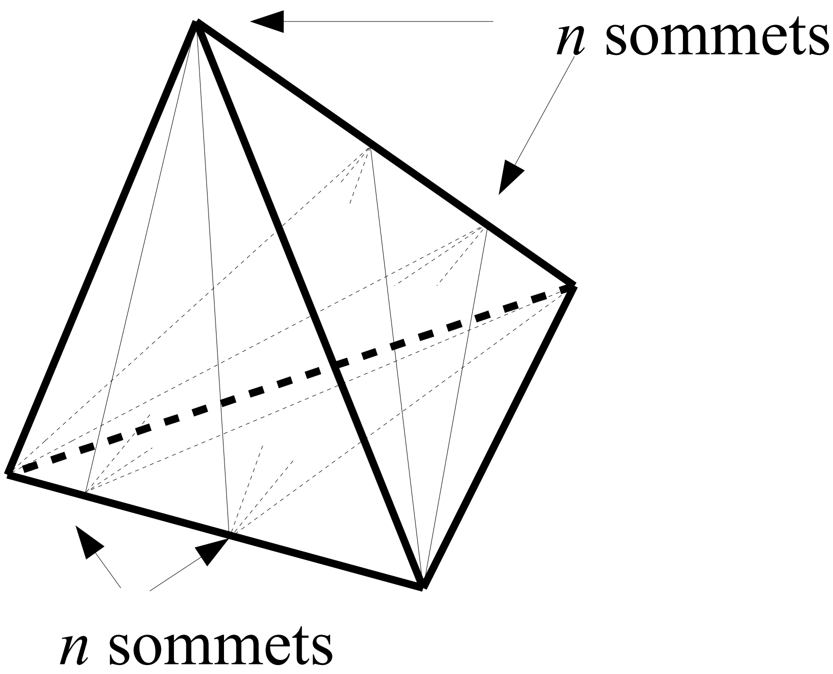
- Quelques remarques
  - Algorithme non optimal
    - Il existe des alternatives (très complexes) en $O(n)$
      - B. Chazelle. Triangulating a simple polygon in linear time. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, IEEE, 1990, pages 220-230.
    - Mais : cet algorithme fonctionne aussi en cas de domaines avec des poches (trous). Dans ce cas il est optimal... la borne inférieure du problème théorique étant précisément $\Omega(n \log n)$

- En 3D : tétraèdrisation d'un polytope à n sommets
  - Beaucoup plus difficile !
  - Sans rajouter de sommets internes (points de Steiner) : pas toujours possible
  - Pour savoir si l'on doit ajouter des sommets internes ou non : problème NP – difficile.

    J Ruppert, R Seidel, On the difficulty of triangulating three-dimensional nonconvex polyhedra, *Discrete & Computational Geometry*, 1992, Springer.
  - Le nombre de sommets internes est borné par $\Theta(n+r^2)$, r est le nombre d'arêtes réflexes (non convexes)
  - Le nombre de tétraèdres n'est pas linéaire avec le nombre de sommets du polyhèdres...

Triangulation de polygones

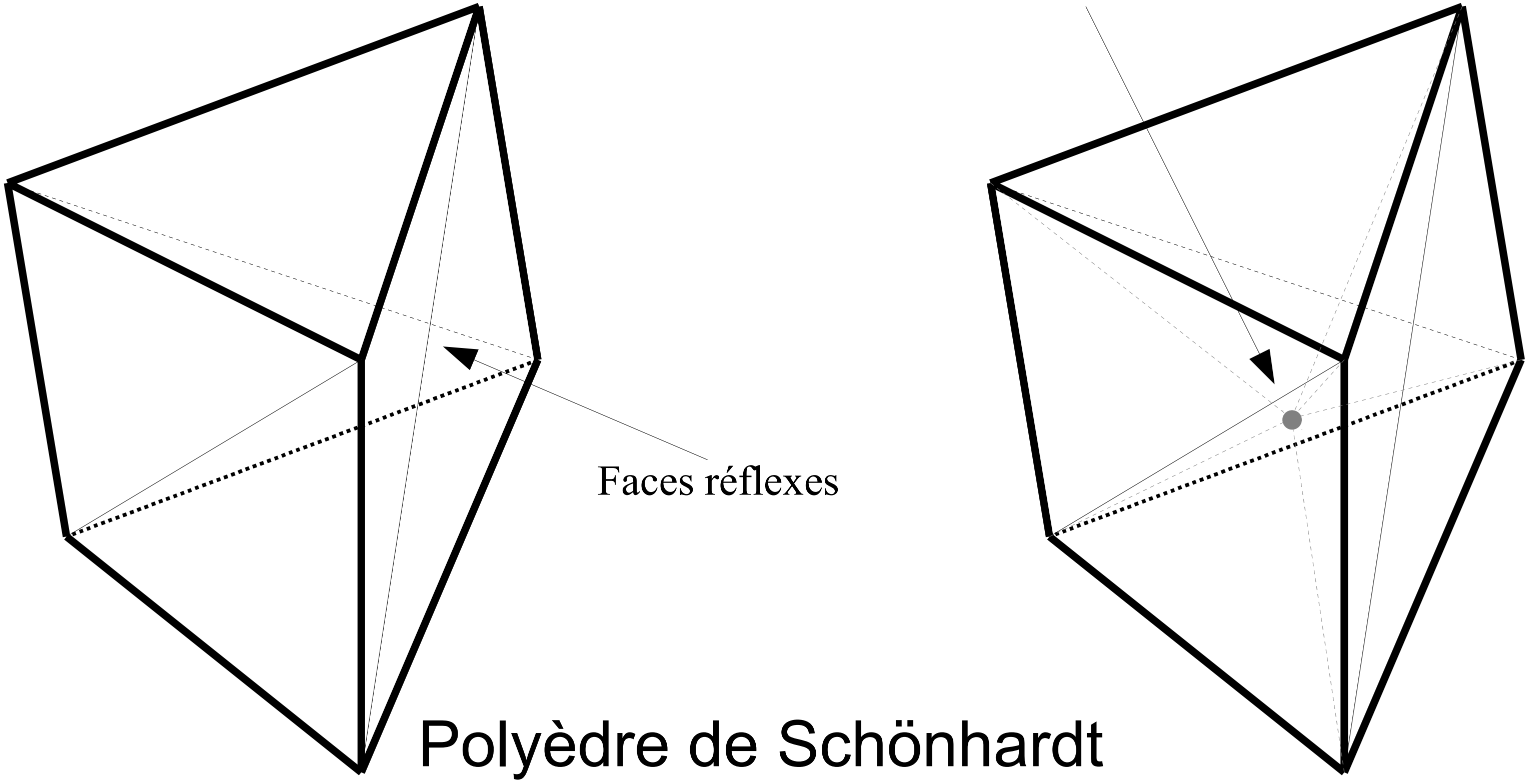
- Exemple de polyèdre simple n'admettant pas une triangulation en $O(n)$...



En tout, il y a $2n$ sommets et $(n-1)^2$ tétraèdres...

Triangulation de polygones

- Exemple de polyèdre simple n'admettant pas de triangulation sans points de Steiner...



Polyèdre de Schönhardt

E. Schönhardt. Über die Zerlegung von Dreieckspolyedern in Tetraeder. *Math. Ann.*, 98:309-312,1928.